OBJECT SEARCH STRATEGY IN TRACKING ALGORITHMS

Dikshya Routray

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela
Rourkela - 769008

Object Search Strategy in Tracking Algorithms

B.Tech Project Thesis

Submitted by:

Dikshya Routray

111EC0185

Supervisor:
Prof. Kamalakanta Mahapatra
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela

Department of Electronics & Communication Engineering,
National Institute of Technology, Rourkela,
Odisha, India – 769 008

CERTIFICATE

This is to certify that the thesis entitled, "**Object Search Strategy in Tracking Algorithms**" submitted by **Dikshya Routray** bearing Roll No. – **111EC0185** in partial fulfilment of the requirements for the award of Bachelor of Technology degree in **Electronics and Communication Engineering** during the session 2014-15 at National Institute of Technology, Rourkela (Deemed University) is an authentic work done by her under my supervision and guidance.

To the best of my knowledge, the matter contained in this thesis has not been submitted to any other university/institute for the award of any Degree/Diploma.

Place: Rourkela
Date:

Prof K.K. Mahapatra
Dept. of Electronics and Communication Engineering,
National Institute of Technology, Rourkela,
769008

ACKNOWLEDGEMENT

I express my gratitude to my thesis guide Prof. Kamalakanta Mahapatra for providing me with this opportunity to work under him and guiding me throughout the whole time-span of the research project. I am indebted to him for helping me to learn the research and writing skills, which have been very beneficial for the current research and will also be for my future career. Without his efforts and patience this research would have never been possible to complete. The concepts, techniques and results presented in this thesis have been guided by him in one or the other way. It has been a great honour and pleasure for me to do research under the supervision of Prof. K.K. Mahapatra. I would like to thank him for being my advisor here at National Institute of Technology, Rourkela.

.

I would like to thank Mr. Vijay Kumar Sharma, Ph. D Scholar under Prof. Kamalakanta Mahapatra, for his indispensable mentoring as and when I needed.

Finally I would like to thank my alma mater National Institute of Technology, Rourkela for giving me the opportunity to study here, enrich my knowledge and hone my skills. It was a great learning experience in the mentally stimulating ambience of the institute.

Dikshya Routray

Contents

ABSTRACT

The demand for real-time video surveillance systems is increasing rapidly. The purpose of these systems includes surveillance as well as monitoring and controlling the events. Today there are several real-time computer vision applications based on image understanding which emulate the human vision and intelligence. These machines include object tracking as their primary task. Object tracking refers to estimating the trajectory of an object of interest in a video. A tracking system works on the principle of video processing algorithms. Video processing includes a huge amount of data to be processed and this fact dictates while implementing the algorithms on any hardware. However, the problems becomes challenging due to unexpected motion of the object, scene appearance change, object appearance change, structures of objects that are not rigid. Besides this full and partial occlusions and motion of the camera also pose challenges.

Current tracking algorithms treat this problem as a classification task and use online learning algorithms to update the object model. Here, we explore the data redundancy in the sampling techniques and develop a highly structured kernel. This kernel acquires a circulant structure which is extremely easy to manipulate. Also, we take it further by using mean shift density algorithm and optical flow by Lucas Kanade method which gives us a heavy improvement in the results.

OBJECT TRACKING: AN OVERVIEW

Tracking is a fundamental problem in computer vision, with applications in video surveillance, human-machine interfaces and robot perception. Real-time computer vision applications like airport safety, road traffic control, video surveillance, robotics, natural human-machine interface, etc. [1], [2] are of great importance today. These applications include machines that can visualize and understand their environment and react according to the perceived parameters and features [2]. This fact signifies the capability of these systems to detect and track objects. Object tracking is thus considered to be the basic task in these kinds of applications [1], [3]. It is a method of detecting the objects moving in a video with respect to time and pursues the objects of interest by estimating the motion parameters [2]. Motion parameters include trajectory, speed and orientation of the object to be tracked [1]. The tracking algorithm as well as the hardware used defines the efficiency of a good tracker.

Usually we make an effort to simplify the algorithms by putting conditional constraints or making some assumptions on the object motion and/or object appearance. Most of the tracking algorithms make an assumption that the object moves smoothly and does not undergo any sudden changes. A very successful approach has been tracking-by-detection. This is due to the powerful discriminative classification in machine learning, and its application to detection with online training.

The current algorithms primarily differ from one another based on their approach to the following questions: How can the object be represented? How can the image features be selected? How are the shape, motion, and appearance of the object modelled? [4] Thus, we find innumerable tracking methods that answer these questions for various situations.

The shape of an object can be represented by (i) Points (ii) Geometric shapes (iii) Object contour and (iv) Skeletal model etc.

For tracking small regions in an image, point representation is used. Point represents the centroid of the object. Sometimes multiple points in an image are also used to represent the shape of an object. In geometric shape representation, object is represented by ellipse and rectangular shapes. Contour representation concentrates on the boundary of an object. Silhouette of an object is considered to be a portion inside it. Geometric shapes can be used to represent both rigid and non-rigid objects. Contour representations are best suited for tracking non-rigid complex objects.

Colour is an important feature mostly used in histogram based object representation. The two important factors which influence the object colour are: the surface reflectance property and the illuminated spectral power density. Colour is represented by three colours i.e. red, green and blue in RGB colour space but HSV colour space which represents hue, saturation and value is more preferred. Edge is used to detect the boundary of an object. This usually plays a significant role in evaluating the image intensities. The edges are not much sensitive to changes in illumination as compared to colour. Hence this is a simple and more accurate method which is used in the places where boundary of the objects is to be tracked.

A general tracking method consists of an appearance model which can evaluate the likelihood that the object of interest is at some particular location; a motion model which relates the locations of the object over time; and a search strategy for finding the most likely location in the current frame. Most object tracking algorithms work on the appearance model may be discriminative or generative or both.

Video analysis consist of three primary steps: detection of objects that are moving called the target objects, tracking of target objects in consecutive frames, and analysis of tracks to study behaviour and motion. In segmentation**,** the image is partitioned into smaller regions. Hence a qualitative segmentation method should be used for successful object tracking.

Figure 1 Uses of Object Tracking

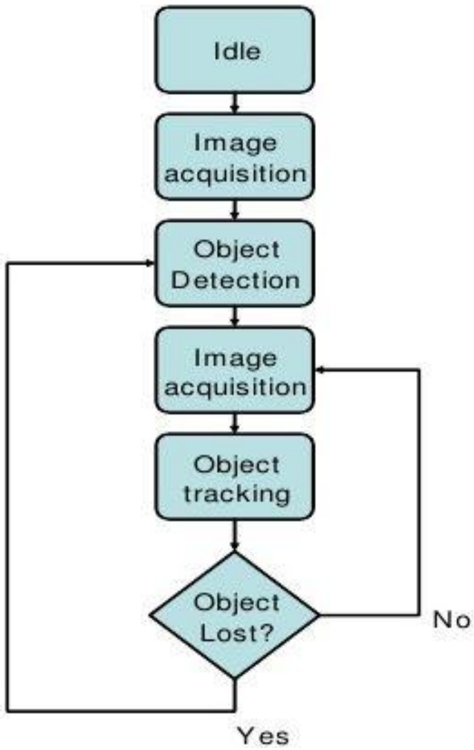


Figure 2 Typical Object Tracking Algorithm

1. LITERATURE SURVEY

The literature on visual object tracking is wide-ranging, and an exhaustive survey is outside the scope of this research. Here, our contributions are focused on the appearance model, as opposed to the motion model and search strategy. Earlier, tracking of point objects using infrared sensors in military applications was only done. Later alpha-beta tracker paved the way for Kalman filter and extended Kalman filters which proved to be very useful for tracking. A typical tracking system consists of three components: (1) an appearance model, which can examine the certainty that the object of interest is at some particular location; (2) a motion model, which relates the locations of the object over time; and (3) an object exploration strategy for finding the most likely location in the current frame. Mostly, contemporary tracking methods emphasize on creating robust appearance models to increasing tracking accuracy, which may be generative or discriminative. A generative method learns an appearance model to represent the target and search for image regions with best matching scores as the results whereas discriminative methods treat tracking as a binary classification problem with local search which estimates decision boundary between an object image patch and the background.

Generative tracking algorithms typically learn a model to represent the target object and then use it to search for the image region with minimal reconstruction error. Black et al. [9] learn a model to represent the object of interest in an offline manner. Nowadays, sparse representation is being used in where an object is modelled by a sparse linear combination of target and trivial templates [17]. Regardless of tremendously showed accomplishment of these online generative tracking calculations, a few issues stay to be comprehended. To start with, various preparing examples edited from successive frames are needed to take in an appearance model. Since there are just a couple of tests at the beginning, most tracking calculations generally assume that the target appearance does not change much amid this period. Notwithstanding, if the presence of the target changes altogether toward the starting, the drift issue is liable to happen. Second, when different examples are drawn at the present target area, it is liable to bring about drift as the appearance model needs to adjust to these possibly misadjusted illustrations. Also, these generative trackers do not use the background information which is likely to improve stability and accuracy of the tracked results.

Moreover, even discriminative classifiers have their own issues. They have to use some efficient feature extraction techniques (e.g., integral image and random projection [6]) that have been proposed for visual tracking. Thus, the number of samples becomes very large, from which features need to be extracted for classification, thereby asking for computationally complex operations. So, both types of models make trade-offs between effectiveness and efficiency. Thus, it remains a daunting task to develop an efficient and robust tracking algorithm.

Tracking may be based on recognition or motion. Tracking in light of recognition is concerned in the recognition of item in progressive pictures and extraction of its position. Its focal point is that it can be attained to in three dimensions and object translation and rotation can be evaluated. At the same time, the issue here is that just perceived objects can be tracked, hence tracking exhibitions are restricted by high computational unpredictability.

Another approach is adaptive tracking-by-detection. As such, the algorithms that use this approach separate the adaptation phase of the tracker into two distinct parts: (i) the generation and labelling of samples; and (ii) the updating of the classifier. While widely used, this separation raises a number of issues. First of all, it is important to design a strategy for generating and labelling samples, and it is not clear how this should be done in a principled manner. The usual approaches rely on predefined rules such as the distance of a sample from the estimated object location to decide whether a sample should be labelled positive or negative. Secondly, the objective for the classifier is to predict the binary label of a sample correctly, while the aim of the tracker is to estimate object location accurately.

Many established learning algorithms such as Boosting [18], Support Vector Machines (SVM) [19], or Random Forests [20] have been used, and are adapted to online training. Contemporary works have focused increasingly on problems of tracking, such as unpredictability in the training labels. Some notable examples use Semi-Supervised Learning and Multiple Instance Learning (MIL Track) to handle this. Going even further, Hare et al. propose Struck, an online version of Structured Output SVM. (cite)

Generally, in visual tracking, a local context consists of a target object and its immediate surrounding background within a determined region. The tracking of an object comprises of two primary steps: representation and localization. The former depends on the modelling of the target while the latter deals with method of target search in subsequent frames. Contours, histograms and feature points help in object representation. Rich textures in the object frames are well represented by using feature points. To understand and get a comprehensive idea about this topic, we studied and implemented four algorithms and improved upon the results of the same. These algorithms, namely Spatio-Temporal Context Learning, Fast Compressive Tracking, Weighted Multiple Instance Learning and Kernelized Correlation Filter (cite) gave a coherent idea to us on their implementation and gave us a perspective on how to improve the accuracy and speed of object tracking. More about these algorithms and implementation will be covered in the next chapter.

2. ANALYSIS AND IMPLEMENTATION OF CONTEMPORARY ALGORITHMS

Apart from a literature survey, it becomes extremely important to study and implement contemporary tracking algorithms so as to understand the nitty gritty of these novel methods. Also, it becomes important to note that certain videos respond well to only a selected few algorithms. Thus, it is essential to conduct a case study first.

Common to most algorithms, a local context consists of a target object and its immediate surrounding background within a determined region.

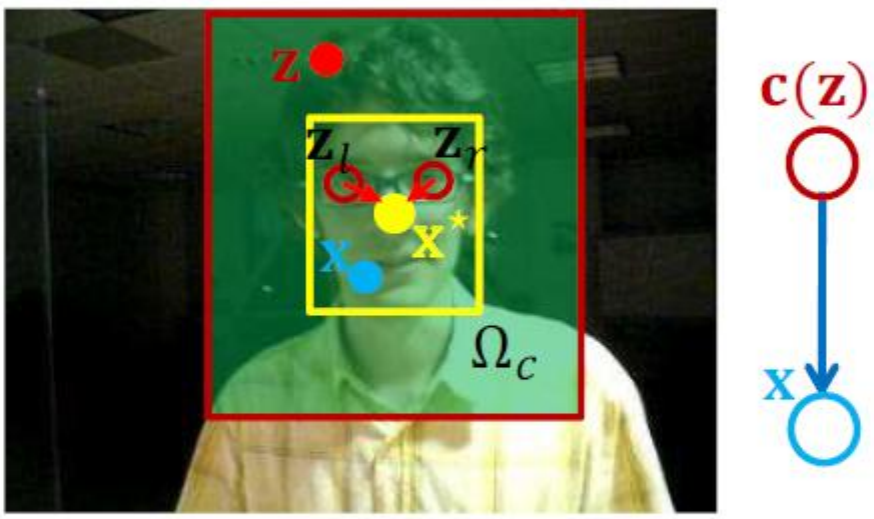


Figure 3. Demarcation of context and target regions by using red and yellow rectangles respectively

The four methods that we have implemented can be enumerated as follows:

- Spatio Temporal Context Learning
- Fast Compressive Tracking
- Online weighted multiple instance learning
- Tracking-by-detection with Kernels

2.1 Spatio Temporal Context Learning

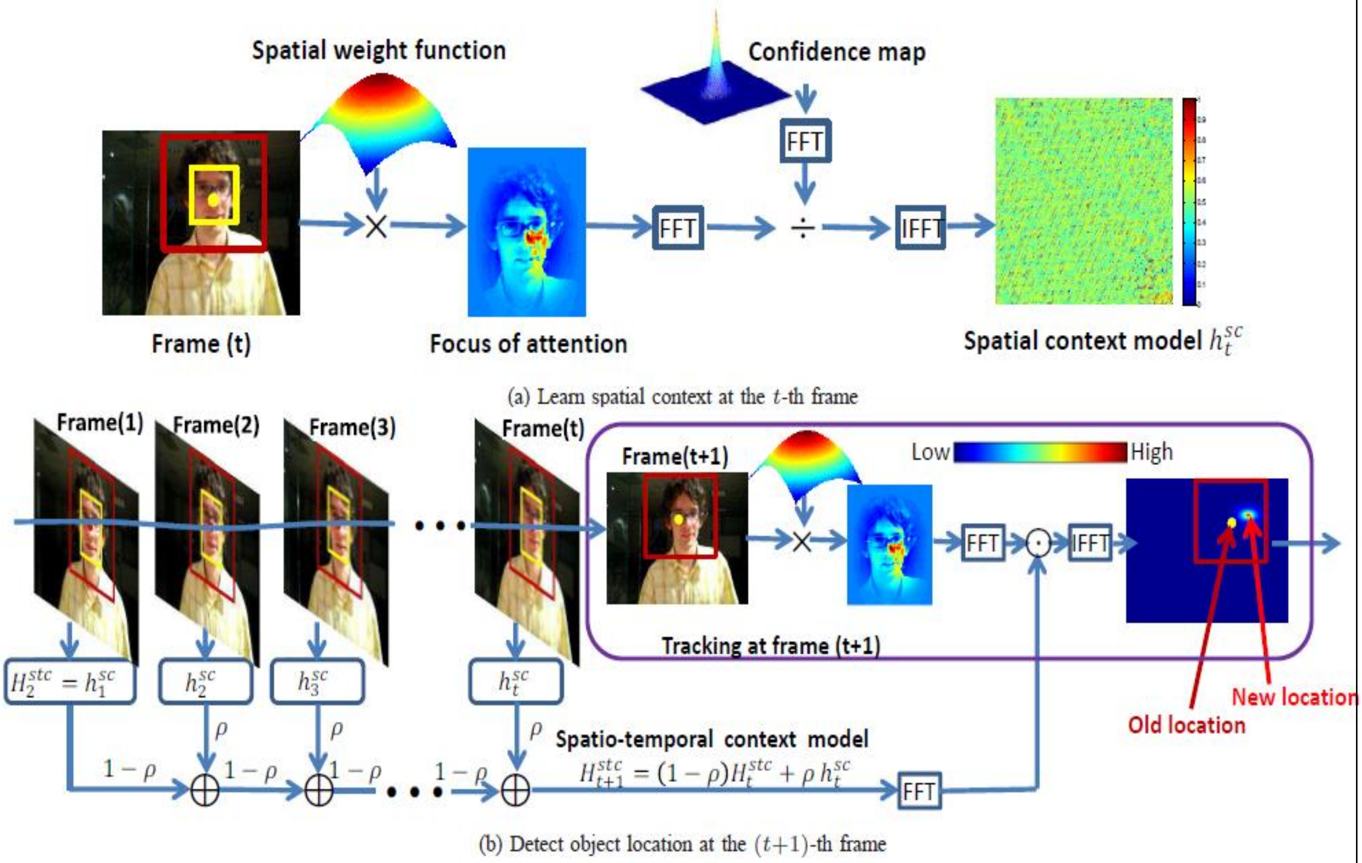


Figure 4 Illustration of algorithmic flow for spatio temporal context learning

This methodology defines the Spatio -temporal connections between the object of interest and its neighbourhood connection in the light of a Bayesian structure, which models the statistical correlation between the low-level feature (i.e., image intensity and position) from the target and its encompassing locales [3]. It is a representation of how our object of interest changes with space and time. The following issue is postured by processing a confidence map, and acquiring the best target area by augmenting an object area probability capacity. The Fast Fourier Transform is received for quick learning and location in this work. To begin with, we take in a spatial context model between the target object and its nearby encompassing foundation in view of their spatial correlations in a scene by taking care of a deconvolution issue. Next, the spatial context model is utilized to overhaul a Spatio -temporal setting model for the following casing. Tracking in the following frame is formed by processing a confidence map as a convolution issue that coordinates the Spatio temporal setting data, and the best object area can be assessed by boosting the confidence map:

$$\begin{aligned} c(x) &= P(x|o) \\ &= \sum_{c(z)\epsilon X^c} P(x, c(z)|o) \\ &= \sum_{c(z)\epsilon X^c} P(x|c(z), o)P(c(z)|o) \end{aligned} \tag{1}$$

Where the conditional probability $P(x|c(z), o)$ models the spatial relationship between the object location and its context information which helps resolve ambiguities when the image measurements allow different interpretations, and $P(c(z)|o)$ is a context prior probability which models appearance of the local context. The proposed tracking algorithm has a learning parameter which takes into account the information gathered from previous frames.

2.2 Real Time Compressive Tracking

This is a powerful and proficient tracking calculation with an appearance model taking into account features extricated in the compressed area [6]. The appearance model is generative as the article can be very much spoken to taking into account the features removed in the compressive space. It is additionally discriminative in light of the fact that we utilize these features to particular the focus from the encompassing foundation by means of a Naive Bayes classifier. The features are chosen by a data safeguarding and non-versatile dimensionality decrease from the multi-scale picture feature space taking into account compressive detecting speculations like arbitrary projections. Thus, a high measurement vector $\mathbb{x}$ to a low dimensional vector $\mathbb{v}$. This lessens computational unpredictability as the matrices are sparse. We accept that the tracking window in the first casing has been dead set. At every frame, we test some positive specimens close to the present target area and negative examples far from the article focus to overhaul the classifier. To foresee the interested object in the following frame, we draw a few examples around the present target area and focus the one with the maximal classification score.

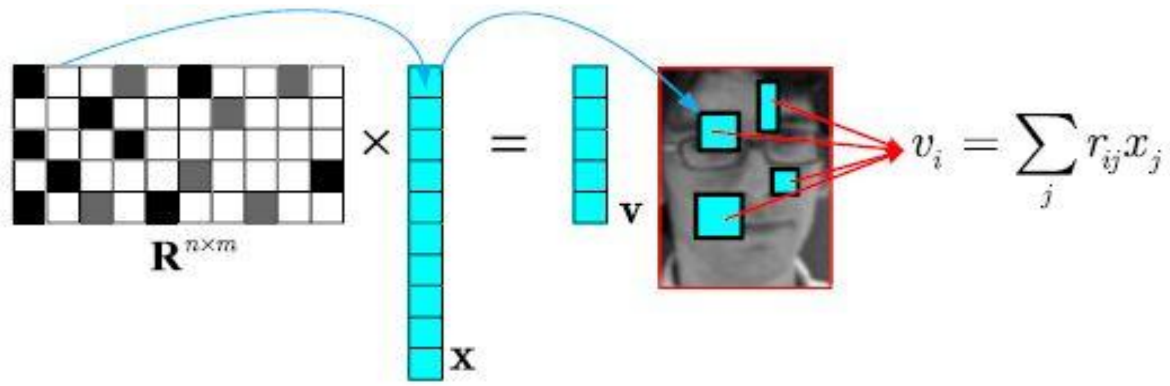


Figure 5 Random Projections for Fast Tracking

2.3 Online Weighted Multiple Instance Learning

There has been much vagueness in picking the positive and negative examples for the classifier. Additionally, utilizing a few positive image patches to redesign a conventional discriminative classifier can make it troublesome for the classifier to take in a tight choice limit. On the off chance that a sack is marked positive it is expected to contain no less than one positive example, else the entire sack is negative [8]. For instance, in the connection of item location, a positive sack could contain a couple of conceivable bounding boxes around every marked article. Consequently, the equivocalness is passed on to the learning calculation, which now needs to make sense of which example in every positive sack is the most right. This calculation additionally incorporates a novel online boosting for MIL.

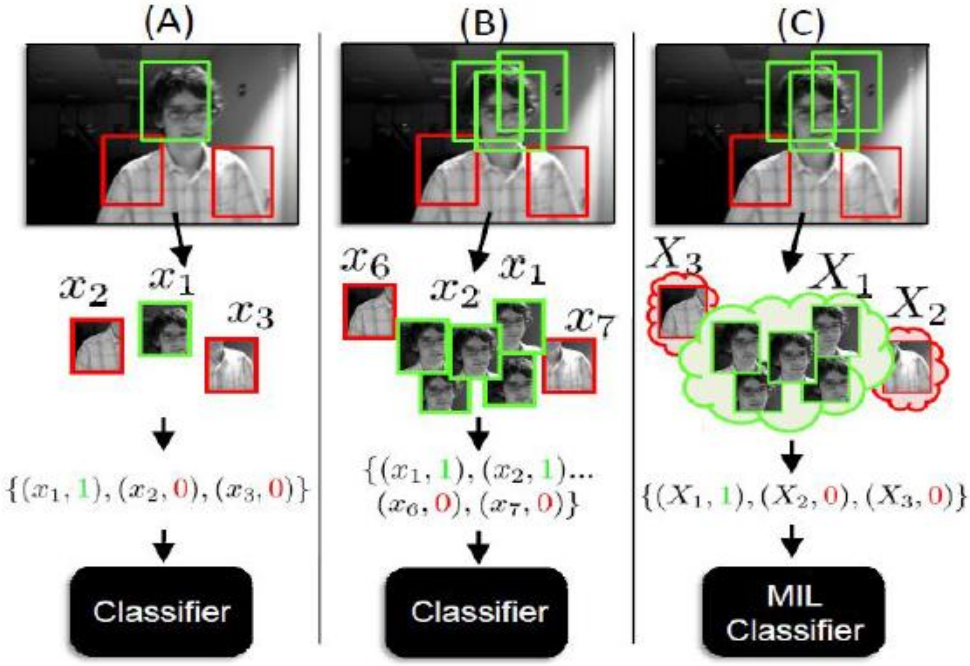


Figure 6 (A)The classifier is updated with a single positive sample (B)The classifier getting updated with several positive samples (C) Proposed Algorithm : All positive samples treated as one bag to make the decision boundary tight

2.4 COMPARISON OF RESULTS OF ABOVE ALGORITHMS

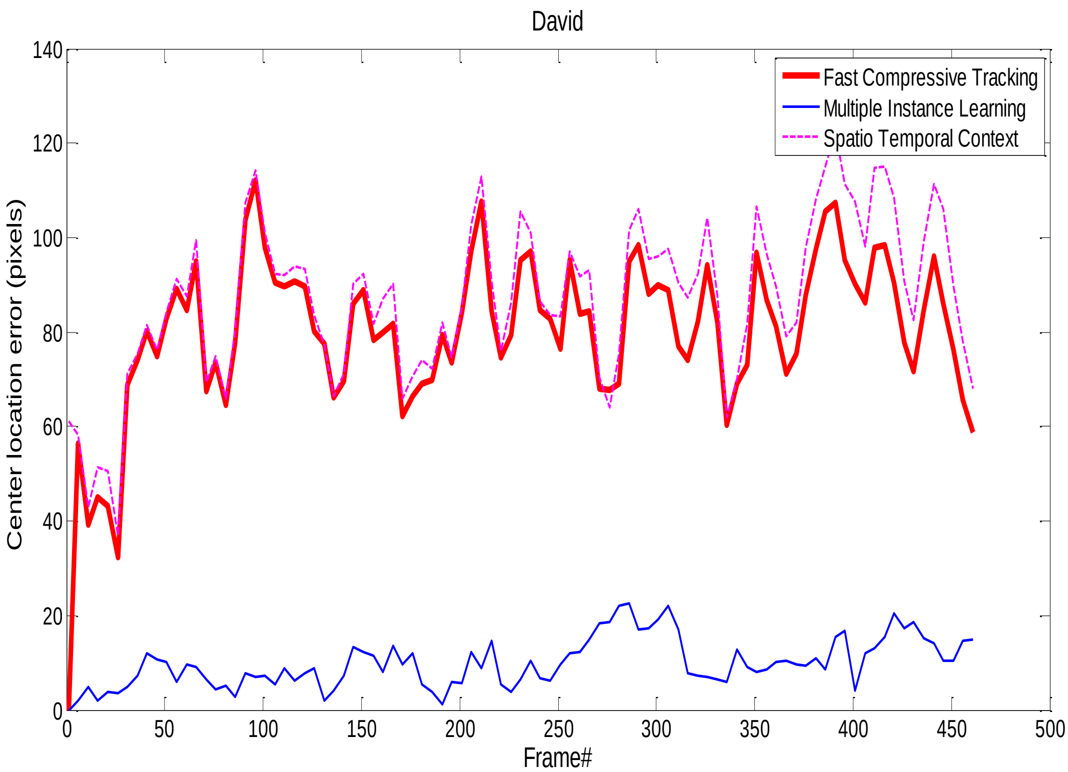
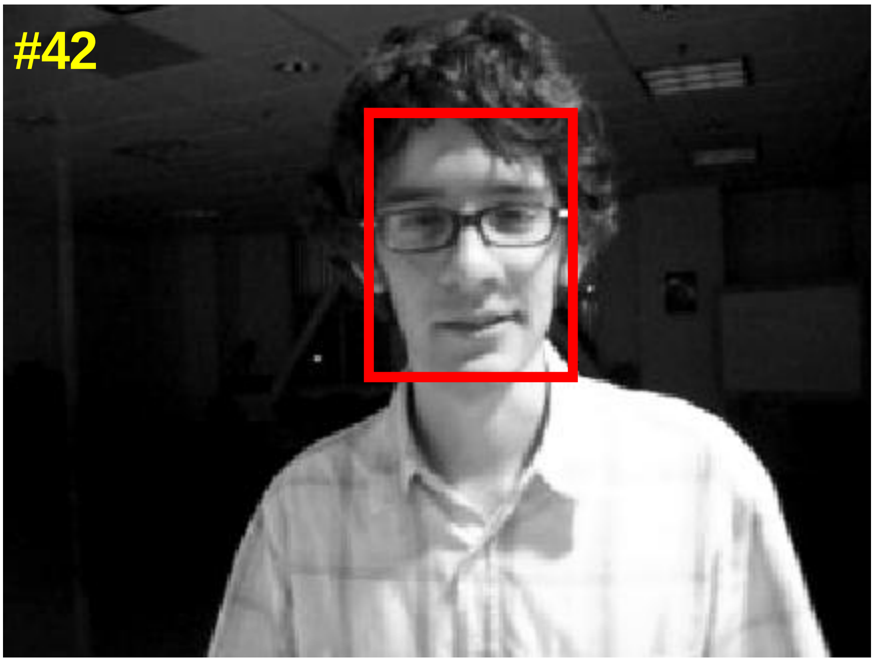


Figure 7 Comparison of center pixel error for above algorithms

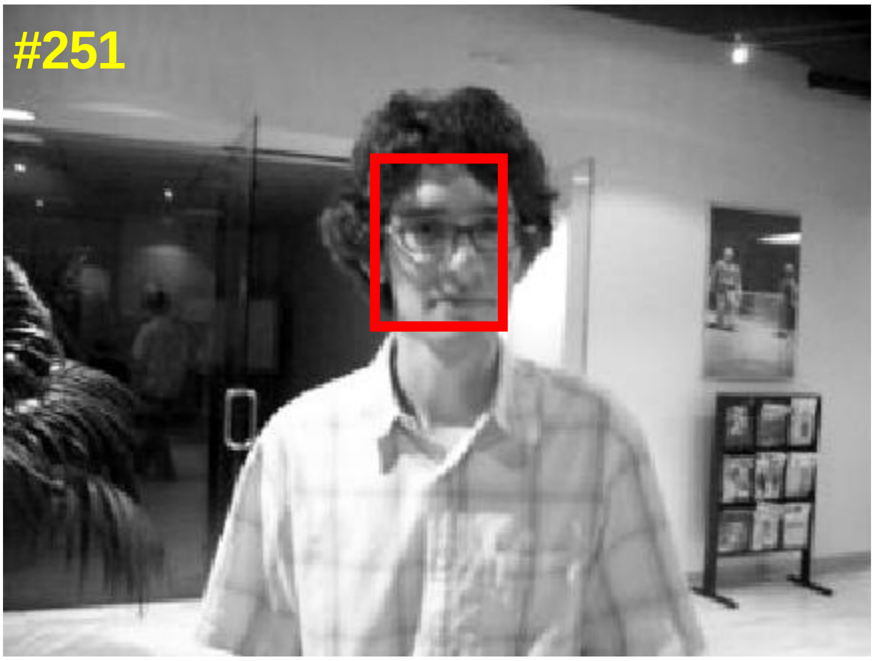
For finding the center pixel error, our database is required to have access to the ground truth which gives the exact positions of the centre pixel of the tracked object. The difference in the tracked centre pixel and its value from the ground truth helps us find center pixel error.

Here, we observe best tracking results in the case of a WMIL Tracker as compared to Spatio Temporal Context Learning and Fast Compressive Tracking for ‘David’.

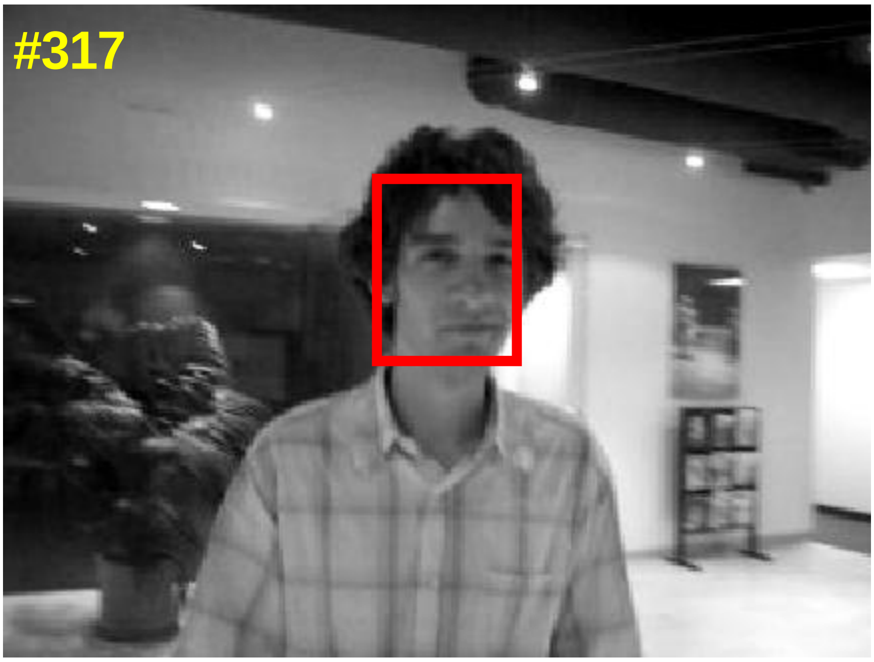


(a)

(b)



(c)



(d)

Figure 8 (a),(b),(c),(d) represent the variation of pose, motion and luminance in successive frames for Spatio Temporal Context Learning

2.5 Tracking-by-detection with Kernels

The majority of the proposed techniques have one thing in like manner: a sparse sampling procedure. In every frame, a few specimens are gathered in the target's neighbourhood, where normally every specimen portrays a sub window. Plainly, there is a considerable measure of excess, since the majority of the examples have a lot of cover. This calculation proposes another hypothetical system to address this. It is demonstrated that the methodology of taking all the sub windows of a picture impels circulant structure [21]. We then make connections to Fourier analysis that permits the utilization of the Fast Fourier Transform (FFT) to rapidly join data from all sub windows, without reiterating over them [14]. It is a hypothetical structure to study nonexclusive classifiers that are prepared with all sub windows (of settled size) of a picture. This is called dense sampling. It demonstrates that the part network for this situation has circulant structure, for unitarily invariant kernels. It gives a shut frame, quick and careful arrangement (all running in O(n^2logn) for nxn pictures) for Kernel Regularized Least Squares with dense sampling (b) Detection at all sub windows with nonexclusive Kernel classifiers. At last, it proposes a tracker taking into account these thoughts which is speedier than existing cutting edge strategies.

Given a set of training patterns and labels $(x_1, y_{1)})$..... $(x_m, y_{m)})$, a classifier $f(x)$ is trained by finding the parameters that minimize the regularized risk. A linear classifier has the form $f(x) =< w, x >$ where <:,:> is the dot product, and the minimization problem is:

$$min_{w,b} \sum_{i=1}^{m} L\big(y_i, f(x_i)\big) + \lambda||w||^2 \qquad (2)$$

Where $L\big(y_i, f(x_i)\big)$ is a loss function, and λ and i varies from 1 to m controls the amount of regularization. This has a closed form solution $\alpha = (K + \lambda I)$ where K is the kernel matrix, I is the identity matrix, and the vector y has elements y_i. It can be proved that by dense sampling that the kernel matrices are circulant. Given a single image x, expressed as a nx1 vector, the samples are defined as $x_i = P^i x$; for all i = 0... n-1 with P the permutation matrix that cyclically shifts vectors by one element.

This algorithm works for non-linear kernels as well.

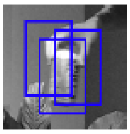
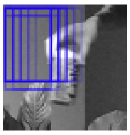
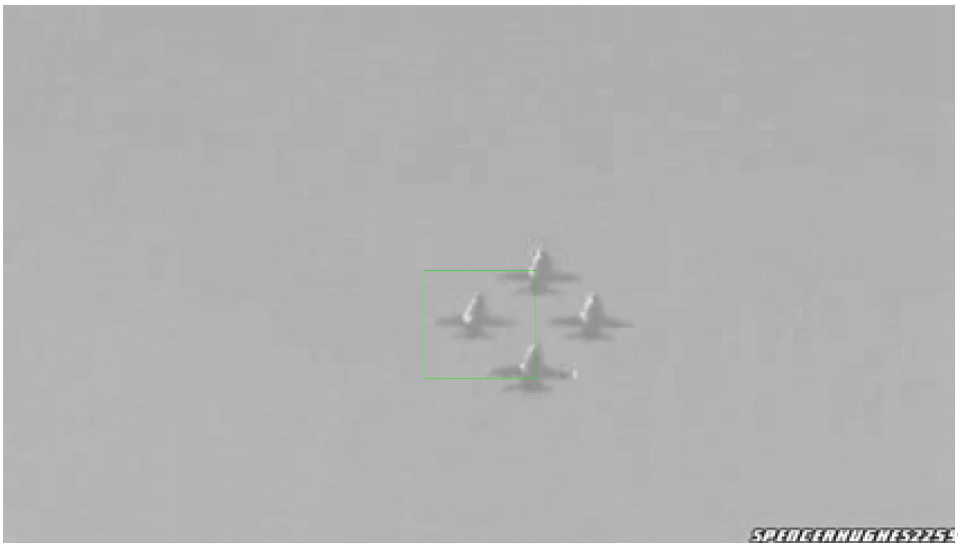
		Storage	Bottleneck	Speed
Random Sampling (p random subwindows)		Features from p subwindows	Learning algorithm (Struct. SVM [4], Boost [3, 6]...)	10 - 25 FPS
Dense Sampling (all subwindows, proposed method)		Features from one image	Fast Fourier Transform	320 FPS

Figure 9 Comparison of performance parameters

(a)

(b)


(c)

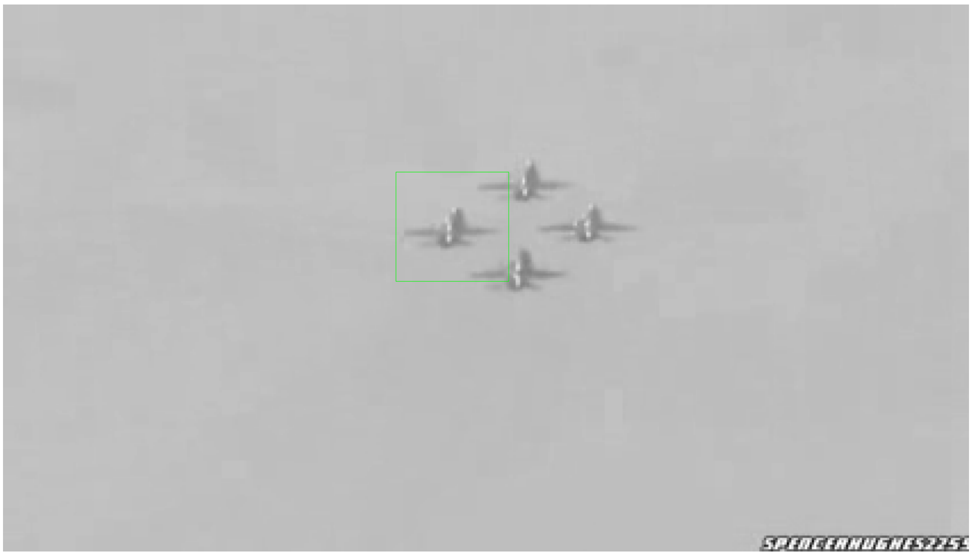

(d)

Figure 10 Performance of kernelized correlation in airshow fleet video

3. TRACKING WITH DENSITY ESTIMATION

3.1 INTRODUCTION

Object tracking is done to recognize moving objects and take after the objects of enthusiasm by assessing the movement parameters [1], [2]. It has extremely pivotal influence in a few computer vision applications [1], [2].These applications incorporate those machines that can gain, handle and comprehend the info images furthermore respond to the environment. In this way the objective of object tracking is to find an object and analyse its orientation. The object to be followed or the target may be of any sort. The object of enthusiasm to be characterized relies on upon the particular application. The targets in building surveillance frameworks may be individual, for some gaming applications the target may be faces or hands. So assessing a few writings and comprehension the issues for tracking, the mean-shift calculation is picked for estimating modes in probability density which is useful for tracking.

3.2 MEAN SHIFT ALGORITHM

Mean-shift algorithm for tracking moving objects was initially given by Comaniciu *et al.*[10]. If we have a set of samples, then according to this algorithm the modes or peaks in a density function is determined. Its applications include segmentation, clustering, tracking, etc. Mean-shift based tracker tracks for a longer time and is more robust as compared to other trackers. This algorithm is basically an iterative process and a non-parametric method [22].

It is basically a tool for finding the modes i.e. peaks in a distribution or a set of data samples the ROI (region of interest). It has the direction same as that of gradient of the density estimate and its size also depends on the gradient. It is computed iteratively for obtaining the maximum density in the local neighbourhood. Near maxima, steps are small and refined. So for mean shift, we take a set of data points, assign the weights, sum them up, divide by the number of weights and then subtract initial estimate from it. Usually the highest mode is taken in a window.

3.3 MEAN SHIFT AND USE IN OBJECT TRACKING

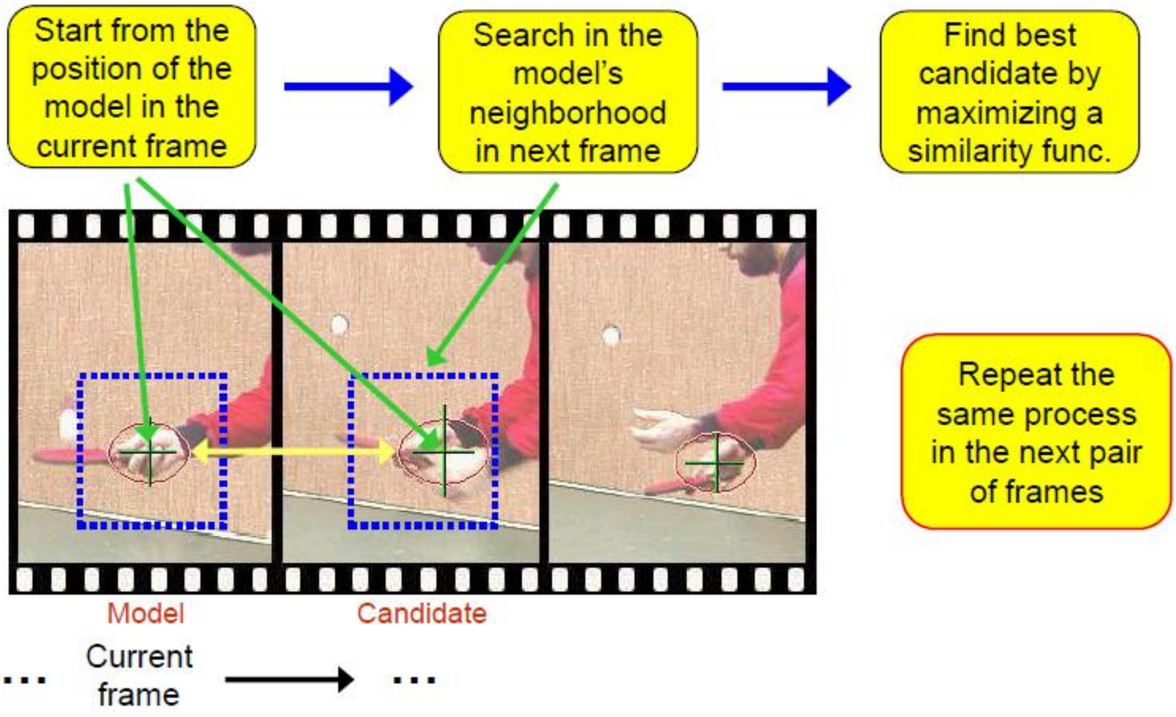


Figure 11 Density Estimation Patch for Mean Shift tracking

Here we use mean shift density estimation along with Kernelized correlation. Therefore, going by the steps, we first initialise the target and estimate a kernel for it. Since, this is the first frame; there is no centre pixel error as we have initialised the target by using the ground truth. Now, when we move on to subsequent frames, we have to compare the density of the targeted patch with the patch from the previous frame. This will give us an estimate of the probability distribution function in the colour space and accordingly, we acquire a mode-seeking similarity function.

Now, we compare this similarity estimate with a threshold value and anything below this threshold is declared dissimilar. However, when we do come across similarity, then we update our mean shift vector and the target position is renewed.

The flow chart below is an indicator of a standard mean shift tracking algorithm which uses a histogram technique.

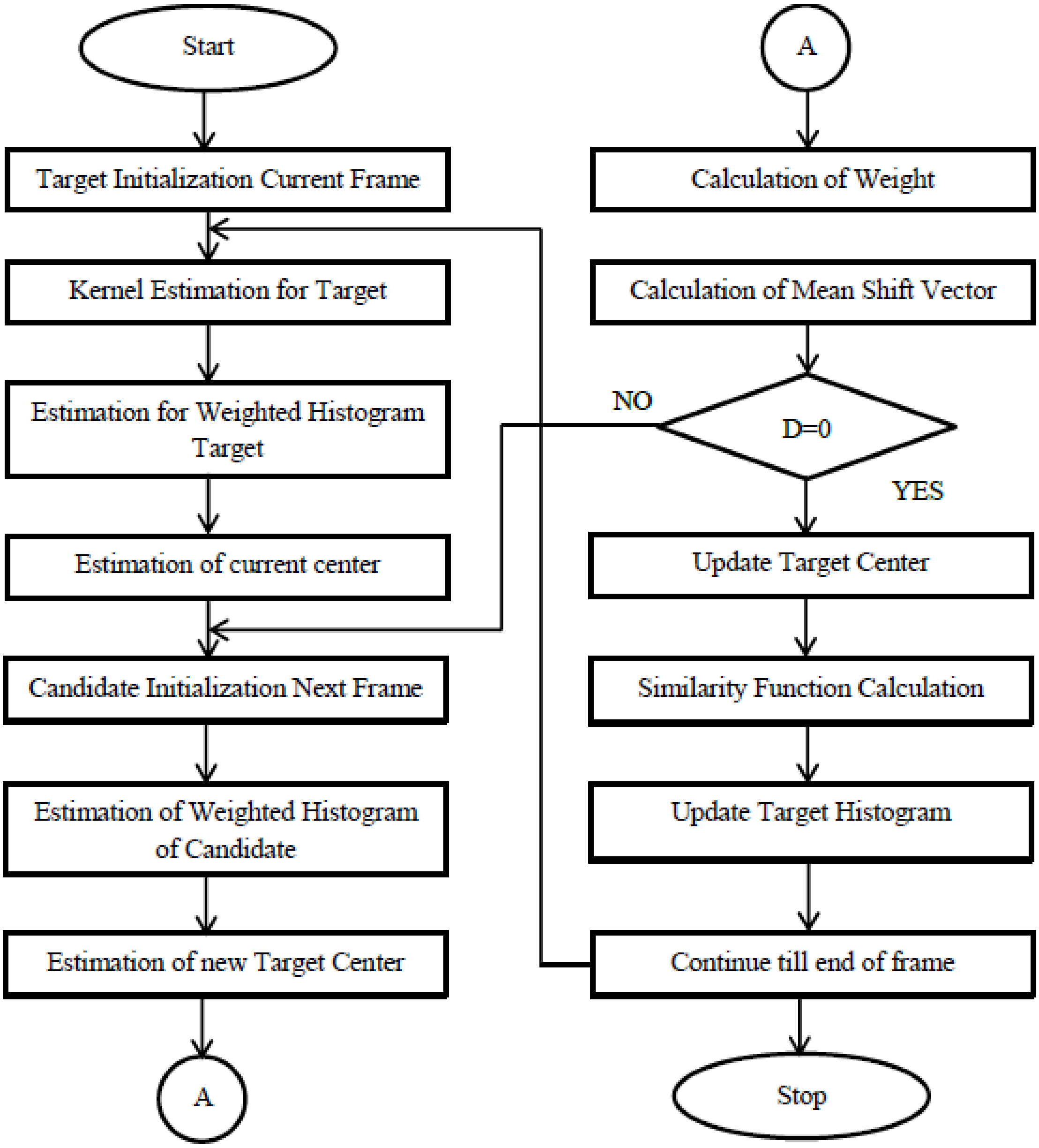


Figure 12 Flow of Algorithm for Mean Shift Tracking using colour histogram

Density estimation is the procedure of assessing the density of the picture. The object of interest has a kernel profile of k. A patch is selected with height H and width W . Two density estimations p and q are assessed with a kernel profile of k. q is the density of reference patch and p is the density estimation of the candidate one. In our case, the MATLAB function we have taken for estimating density is as follows :

```
function [f,w] = Simil_func(q,p,T2,k,H,W)

w = zeros(H,W);
f = 0;
for i=1:H
    for j=1:W
        w(i,j) = sqrt(q(T2(i,j)+1)/p(T2(i,j)+1));
        f = f+w(i,j)*k(i,j);
    end
end
```

w(i,j) is the weight factor that will ultimately give us an estimate of the similarity.

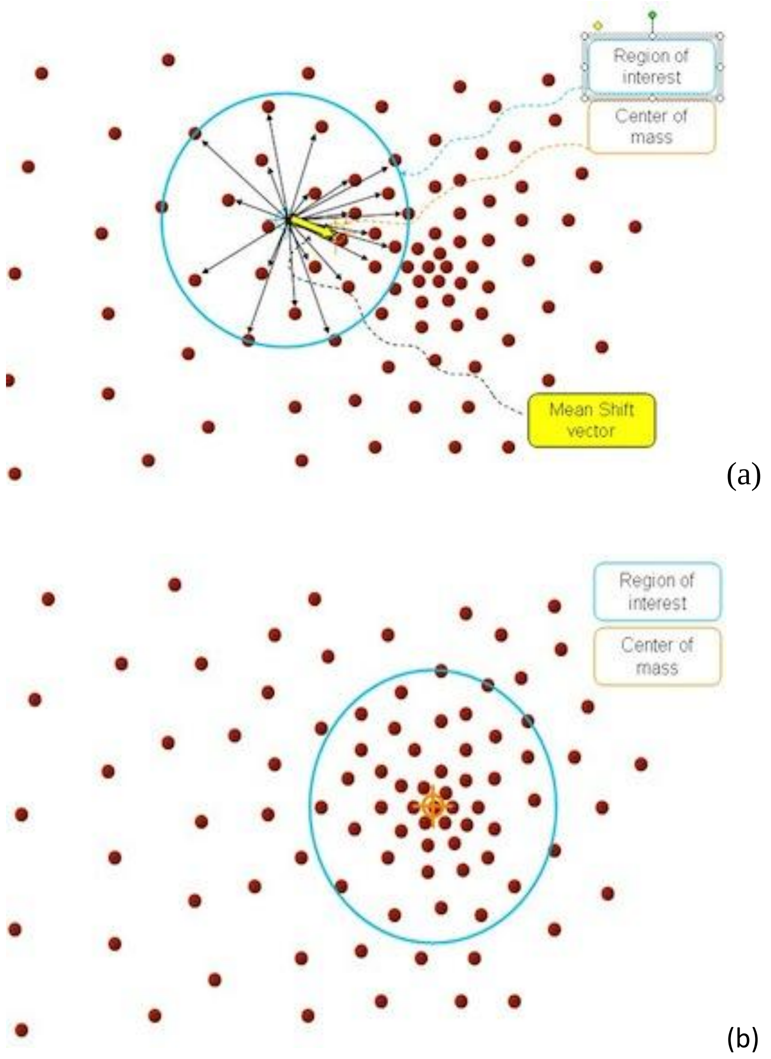


Figure 13 (a) & (b) Pictorial depiction of Mean Shift Vector by directing flow towards Centre of Mass

(b)

3.4 RESULTS:

This proposed algorithm on density estimation was tried on different videos and few results are shown here

A. TIGER

Figure 14 Tracking of tiger using Density Estimation Algorithm despite occlusion and change in appearance

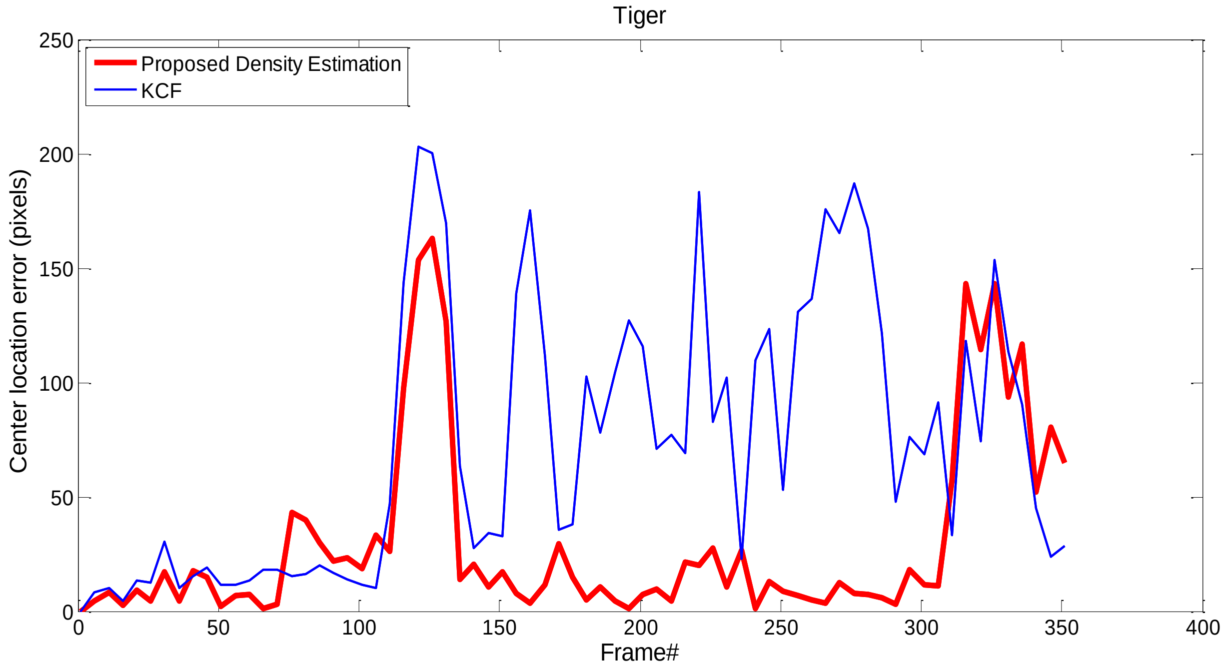


Figure 15 Centre Pixel Error Comparison for KCF and KCF with density estimation

B. SHAKING

Figure 16 Tracking in video 'shaking'

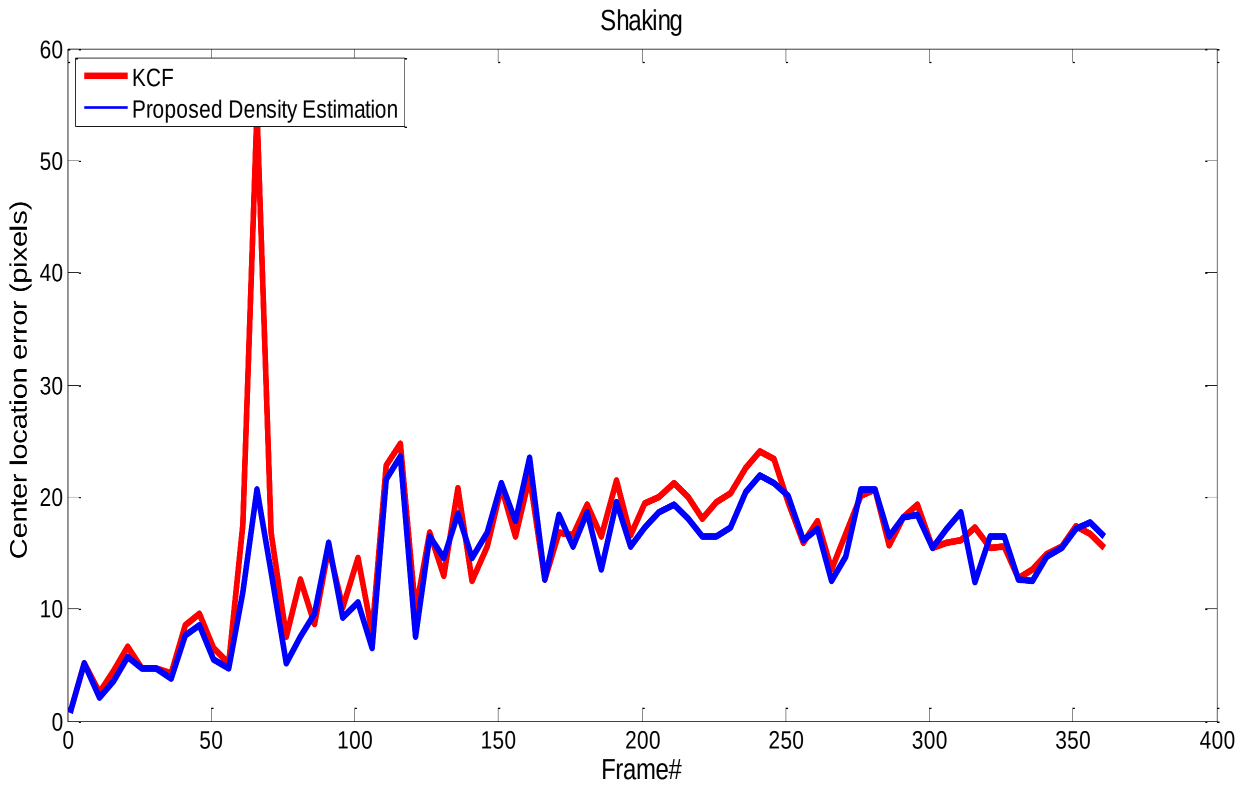


Figure 17 Comparison of centre pixel error for KCF and KCF using Density estimation where the results are somewhat more comparable

In both the video results above, we notice how significant reduction can be noticed in the centre pixel error when density estimation is used.

3.5 CONCLUSION:

The target localisation is done by performing tracking through mean shift method iteratively. This algorithm using mean shift estimates probability density of the colour space and has been found to perform spectacularly with the use of Kernelized Correlation Filter. But a disadvantage of this method is that when the object has a non rigid structure and its shape and size changes, the performance deteriorates. Also, since we take a PDF in the colour space, it is only possible to track RGB images and not greyscale ones, which is a major demerit.

4. TRACKING WITH VELOCITY ESTIMATION

Intuitively, it is expected that if the object moves around in the image plane, its velocity is bound to change from one frame to another. If the velocity changes slowly in consecutive frames, it is possible to track the given object. This can be done proficiently by using an algorithm called optical flow which calculates the motion between two image frames taken at two time intervals with very less time difference**.**

4.1 QUIVER PLOT

A quiver plot displays velocity vectors as arrows with components (u,v) at the points (x,y). Thus, we will be representing the velocity estimations in frames using quiver plots. The figure below can serve as an example of a quiver plot, where the velocity at the pixels is represented using indicative arrows.

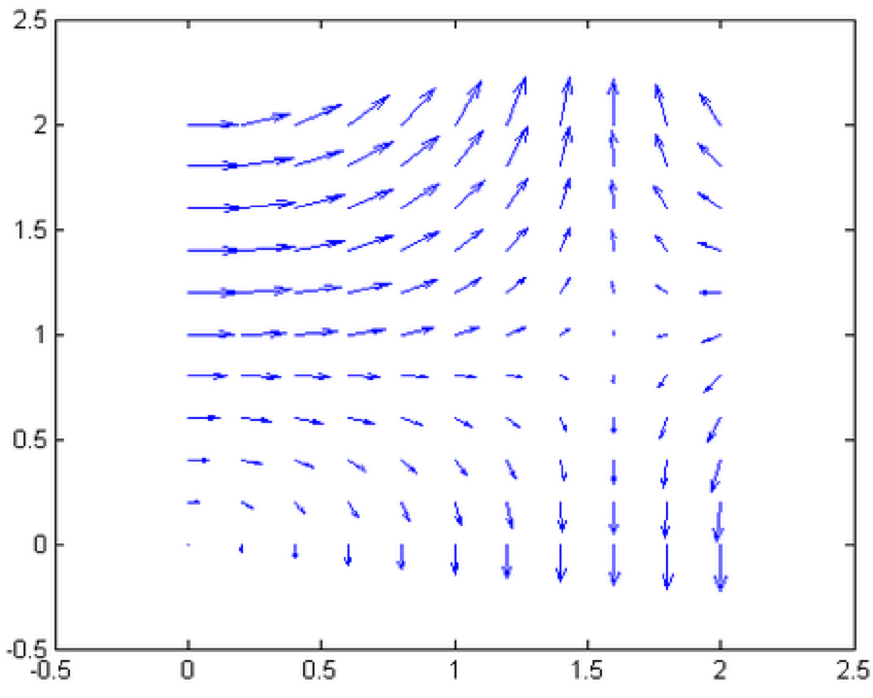


Figure 18 Illustration of Quiver Plot

The two most preferred methods for calculation of optical flow are: Horn & Schunck Optical Flow Method[7] and Lucas & Kanade Tracking Method [5]. In our proposed algorithm which aims to improve the results of Kernelized Correlation Filter uses the L&K Method.

A few assumptions in this method:

- The motion between the frames is small.
- The brightness is assumed to be constant.
- The second order truncation of the Taylor Series expansion is valid.

4.2 APPLICATIONS

Applications of optical flow include motion based segmentation, extracting a structure from motion (3D shape and Motion) and alignment (Global motion compensation), camcorder video stabilization, UAV video analysis and video compression.

4.3 LUCAS & KANADE ALGORITHM FOR OPTICAL FLOW ESTIMATION

The optical flow estimation comprises in the calculation of the motion field that depicts the pixel relocations between progressive frames, which is characterized for every pixel as a velocity vector (u, v). Let the force of the pixel in the (x, y) position in the picture at time t, and the pixel $(x + u, y + v)$ in the picture of time t + 1 does not change. As remarked anytime recently, this is just a legitimate suspicion for moderate moving objects between frames (it relies on upon the separation from the object to the camera, on the 3D object velocity).

$$I(x, y, t) = I(x + u, y + v, t + 1) \qquad (3)$$

This comparison is unravelled by linearization utilizing the first order Taylor expansion and this prompts the ensuing:

$$I_x u + I_y v + I_t = 0 \qquad (4)$$

where the subscripts depict the partial derivatives in every axis (x, y, and t). Because of the way that from (2), we can't find a unique solution, we require another equation to discover the solution of the optical flow estimation (just velocity vectors). This can be done in various ways. Subsequently, we process the movement estimation for a pixel concentrating just on its neighbourhood and disregarding the rest and it is bolstered in light of the fact that nearby pixels are prone to compare to the same objects and, hence, comparable velocity vector qualities can be expected. This methodology was firstly proposed by Lucas & Kanade [5]. The L&K methodology is a standout amongst the most exact, computationally effective, and generally utilized techniques for the optical flow calculation. It is a local technique that has a place with the gradient based ones, on the grounds that its processing is taking into account the image derivatives and on the grounds that it illuminates the past mathematical statement by accepting that the flow is steady in the same neighbourhood. This is the model which we have chosen for our usage. Also, we appraise the optical flow as the quality which minimizes the energy capacity constructing an over-constrained comparison framework

$$E(u,v) = \frac{1}{2}\sum_{i\epsilon\Omega}(W_i^2(I_x u + I_y v + I_t)^2) \qquad (5)$$

where W_i stands for the weight matrix of the pixels in neighbourhood. And then, for the resolution of the system of equations, we use a least squares-fitting procedure.

$$(u,v) = (A^T W^2 A)^{-1} A^T W^2 b \qquad (6)$$

Thus, the component of the flow in the gradient direction was already determined and the component of the flow parallel to an edge is unknown for which the least squares solution is found by pseudo inverse method.

$$A^T W^2 b = \begin{bmatrix} \sum_{i\epsilon\Omega} W_i^2 I_{xi} I_{ti} \\ \sum_{i\epsilon\Omega} W_i^2 I_{yi} I_{ti} \end{bmatrix} \qquad (7)$$

$$A^TW^2A = \begin{bmatrix} \sum_{i\epsilon\Omega} W_i^2 I_{xi}^2 & \sum_{i\epsilon\Omega} W_i^2 I_{xi} I_{ti} \\ \sum_{i\epsilon\Omega} W_i^2 I_{yi} I_{ti} & \sum_{i\epsilon\Omega} W_i^2 I_{yi}^2 \end{bmatrix} \qquad (8)$$

RESULTS

A. VELOCITY ESTIMATION IN AIR SHOW FLEET

Figure 19 Tracking a certain plane in the air show fleet

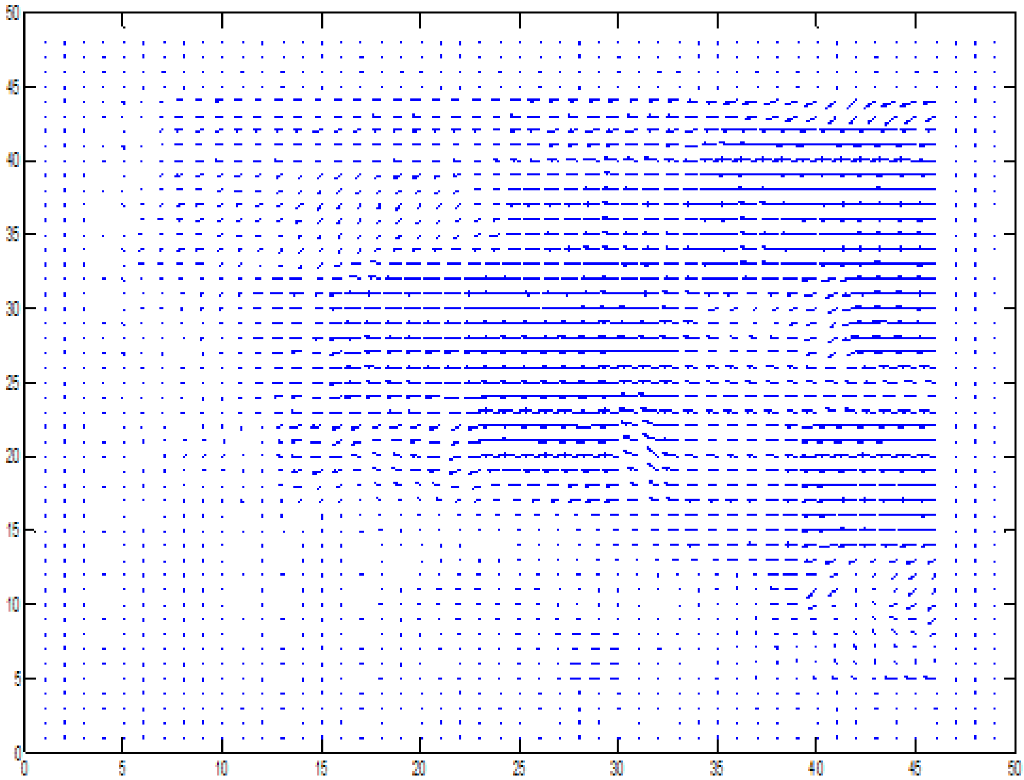


Figure 20 Velocity estimation of air show fleet

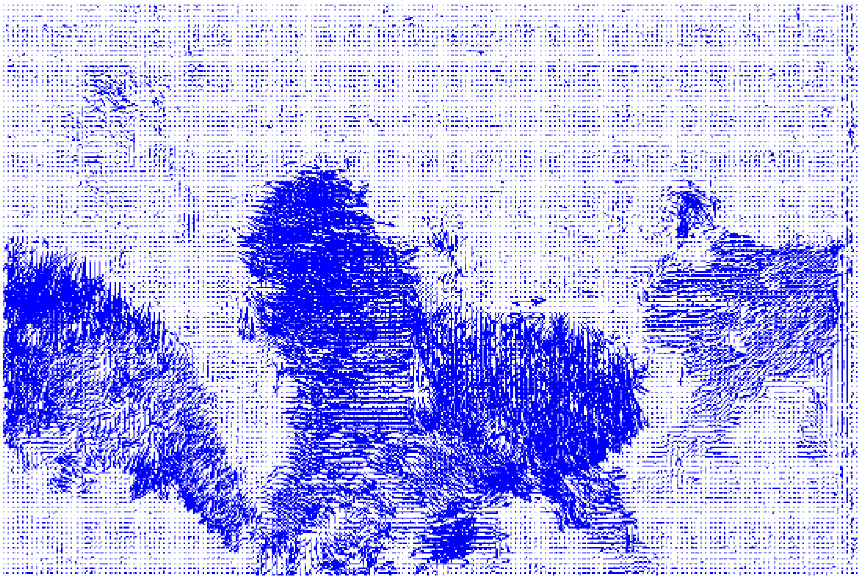


Figure 21

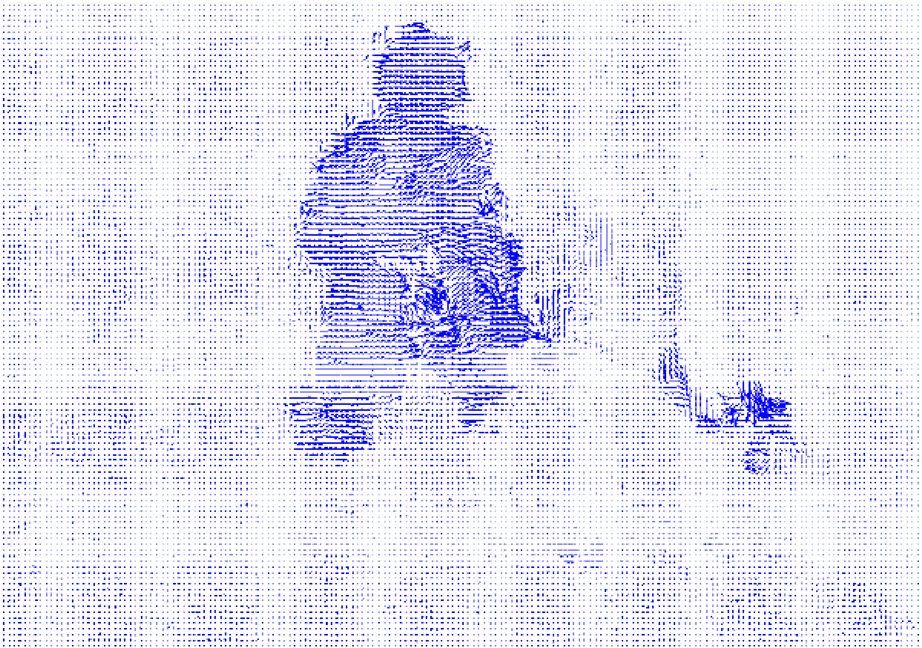


Figure 22 & 21 Velocity Estimations in the above images

B. TRACKING RESULTS AND ERRORS

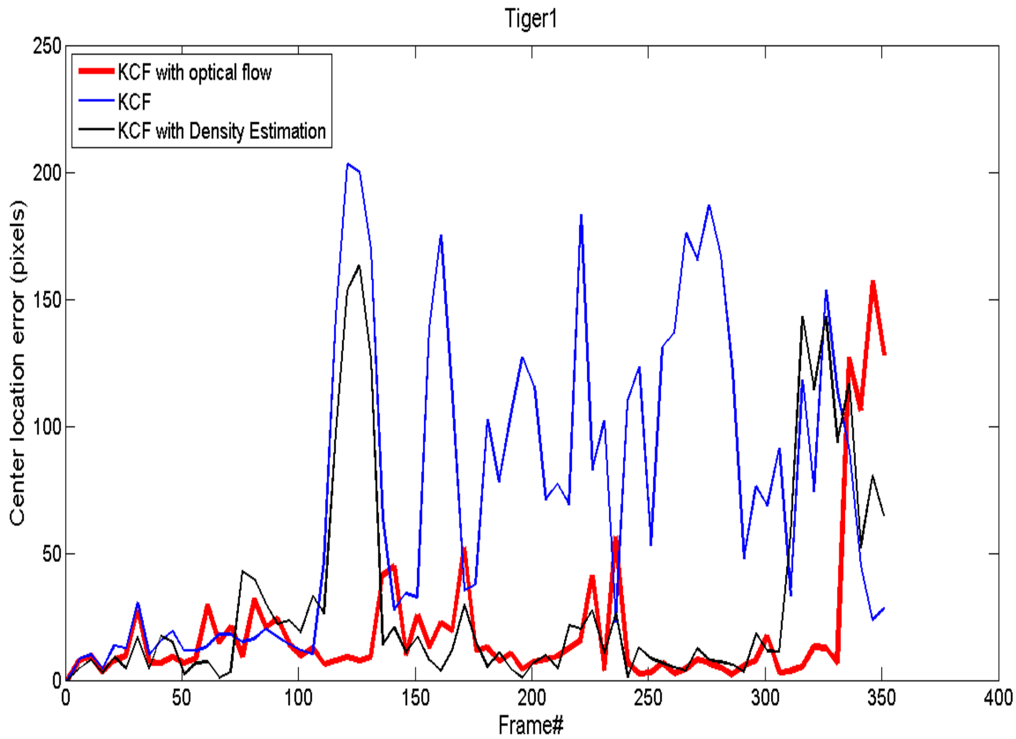


Figure 23 Centre Pixel Error for 'tiger1' video

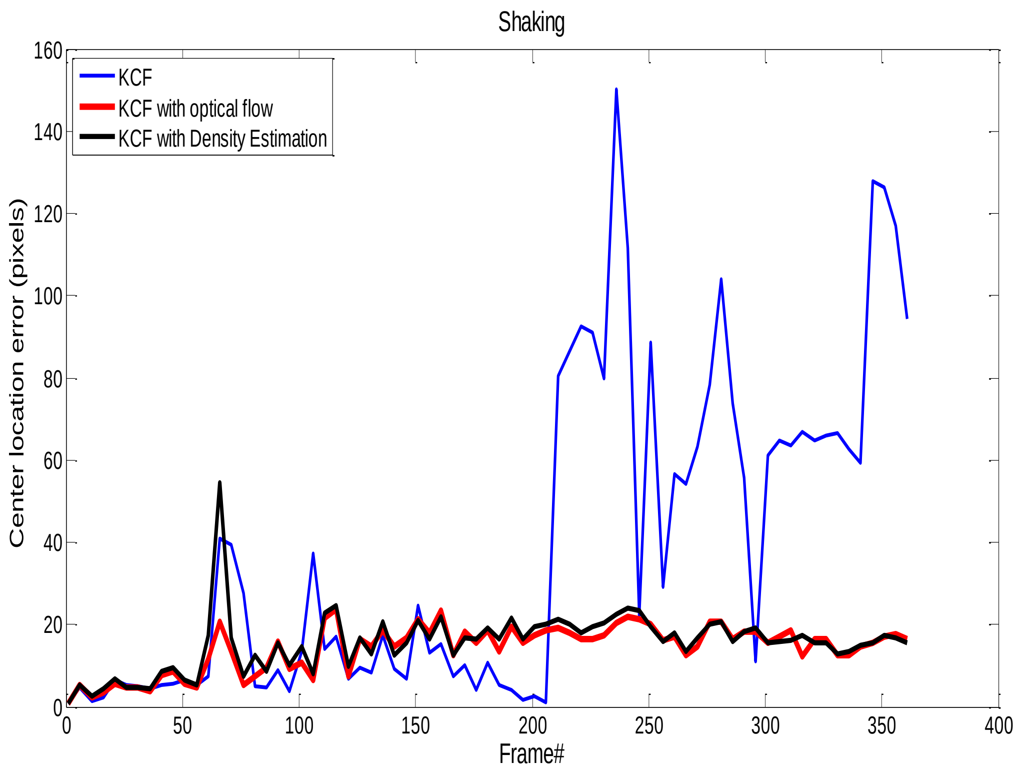


Figure 24 Comparison of Centre Pixel Error for 'shaking' video

4.4 CONCLUSION

Optical Flow using Lucas and Kanade method helps mitigate the aperture problem and gives a good estimate of the horizontal component of velocity. It is extremely useful for edge detection and improving object tracking results with kernel profile k. The improvements we notice are huge and sure are good contributions to contemporary object tracking methods.

5. CONCLUSION AND FUTURE WORK

An object tracking problem has to focus on several salient features:

- How robust the algorithm is (by estimating centre pixel error, tracking failures etc.);
- What are the real time computational processing requirements in terms of space and time complexities, memory requirements for frame extraction, time taken by the algorithms to converge for minimal error in tracking;
- What is the error tolerance and laxity allowed for the algorithm;
- What is the feasibility of implementation of the algorithm on hardware so as build a tangible product.

An extensive literature survey brought into account many algorithms which had focussed on the appearance models for object detection and took all the above salient points into consideration. We mostly focused on Kernelized Correlation and improved its results significantly by using density estimation and optical flow.

Object detection is a promising field of study in computer vision and has a wide ranging future scope. For this particular work of research, the next step lies in hardware implementation. This has become extremely important because many algorithms require more sophisticated improvements when it comes to testing their tangibility. Also, while improving accuracy, we had to compromise on speed of tracking, which must be taken care of when working in future.

7.REFERENCES

[1] Gian Luca Foresti,"Object Recognition and Tracking for Remote Video Surveillance", *IEEE Trans. on Circuits Syst. Video Tech.*, vol. 9, pp. 1045-1062, Oct. 1999.

[2] Emilio Maggio and Andrea Cavallaro, *Video Tracking*, 1st ed., John Wiley and Sons, United Kingdom: Wiley, 2011.

[3] K Zhang, L Zhang and M HYang, "Fast Tracking via Spatio Temporal Context Learning," arXiv preprint arXiv:1311.1939, 2013

[4] Alper Yilmaz , Omar Javed, "*Object Tracking: A Survey*", ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication date: December 2006. pp. 1-45

[5] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," pp. 674 – 679, 1981.

[6]] Kaihua Zhang, Lei Zhang and Ming-Hsuan Yang, "Fast Compressive Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence., vol. 36, no. 10 , October 2014.

[7] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185 – 203, 1981.

[8] B. Babenko, M H Yang, S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," IEEE Trans. Pattern Analysis and Machine Intelligence., vol. 33, no. 8 , December, 2010.

[9] Black, M., Jepson, A.: Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation. IJCV 38, 63–84 (1998)

[10] D.Comaniciu, V.Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean-shift", *IEEE Proc.Comput. Vis. Pattern Recog*., pp. 673-678, 2000

[11] D.Comaniciu, V.Ramesh and P. Meer, "A robust approach towards feature space analysis", *IEEE Trans. Pattern Anal.Machine Intell*., vol.24, pp. 603-619, 2002.

[12] J.F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multiobject tracking with merged measurements. In ICCV, 2011.

[13] D.Comaniciu, V.Ramesh and P. Meer,"Kernel-based object tracking", *IEEE Trans.Pattern Anal. Mach. Intell*., vol.25, pp. 564-577, 2003.

[14] Hare, S., Saffari, A., Torr, P.: Struck: structured output tracking with kernels. In: ICCV, pp. 263–270 (2011).

[15] R C Gonzalez and R. E. Woods. Digital image processing. Prentice Hall , 2008.

[16] K. Zhang , L. Zhang , "Real-Time Compressive Tracking" ECCV 2012, Part III, LNCS 7574, pp. 866879, 2012. Springer-Verlag Berlin Heidelberg 2012.

[17] Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. PAMI 33, 2259–2272 (2011)

[18]H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In ECCV , 2008.

[19] S. Avidan. Support vector tracking. TPAMI, 26(8):1064{1072, 2004.

[20]A. Safari, C. Leistner, J. Santner, M. Godec, and H. Bischof. Online random forests. In 3rd IEEE ICCV Workshop on On-line Computer Vision , 2009.

[21] J. F. Henriques, R. Caseiro, P Martins, and J Batista,“Exploiting the Circulant Structure of Tracking-by detection with Kernels”

[22] Jaideep Jeyakar, R.Venkatesh Babu and K.R. Ramakrishnan, “Robust object tracking with background weighted local kernels”,*Comput. Vis. Image Understanding*, vol.112,pp.296-309,2008.